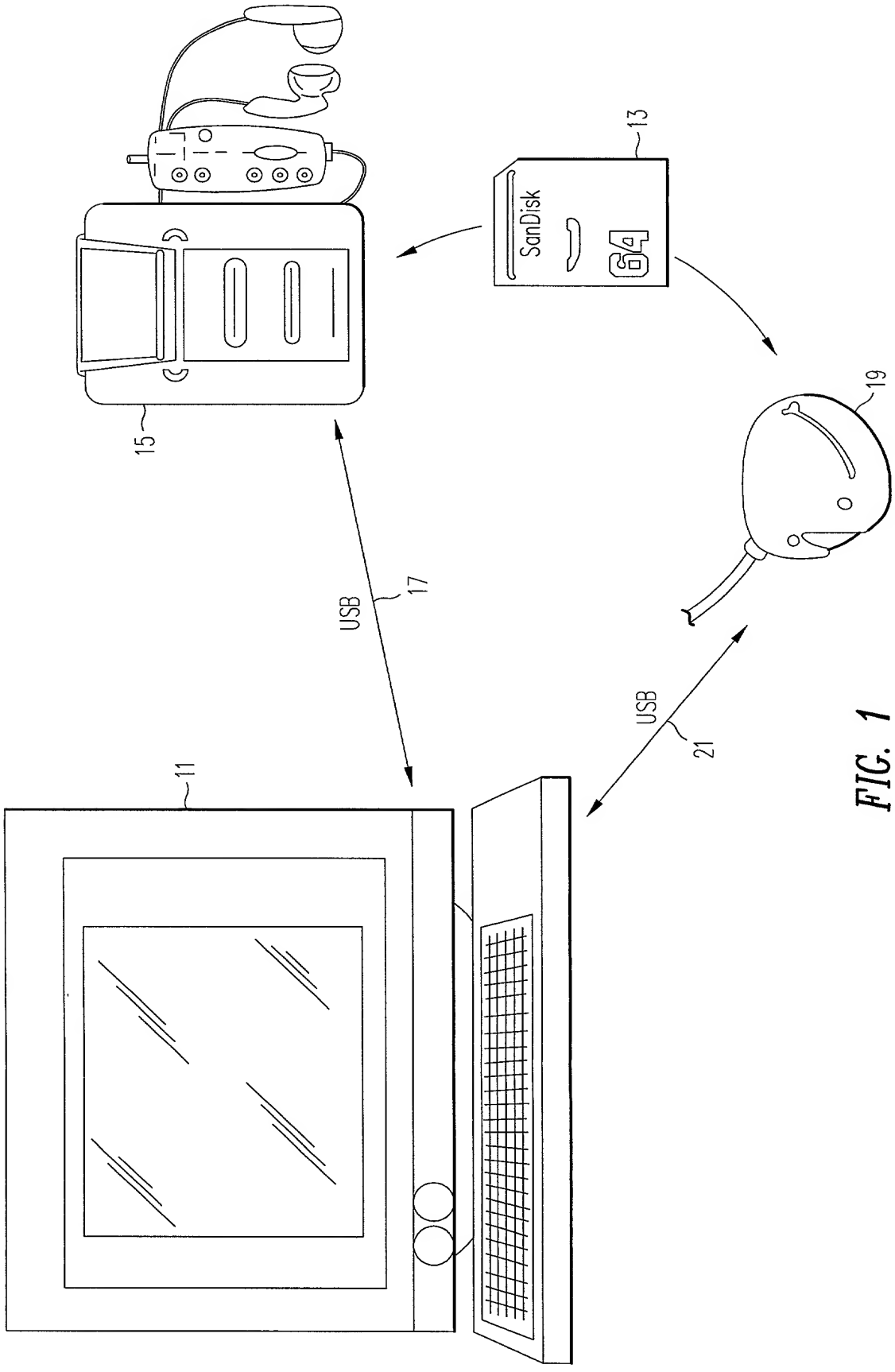


FIG. 1

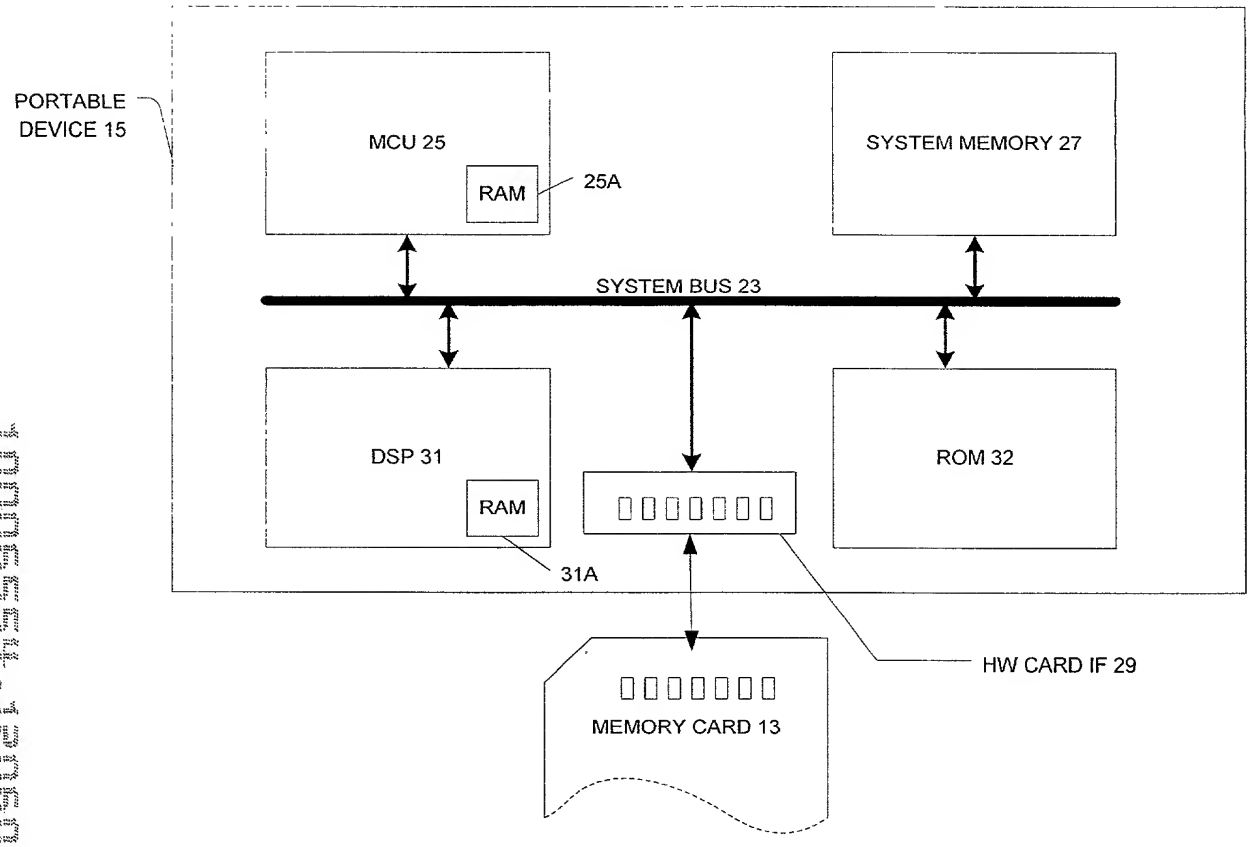

PORTABLE
DEVICE 15
MCU 25
RAM
25A
SYSTEM MEMORY 27
SYSTEM BUS 23
DSP 31
RAM
31A
ROM 32
HW CARD IF 29
MEMORY CARD 13

FIG. 2

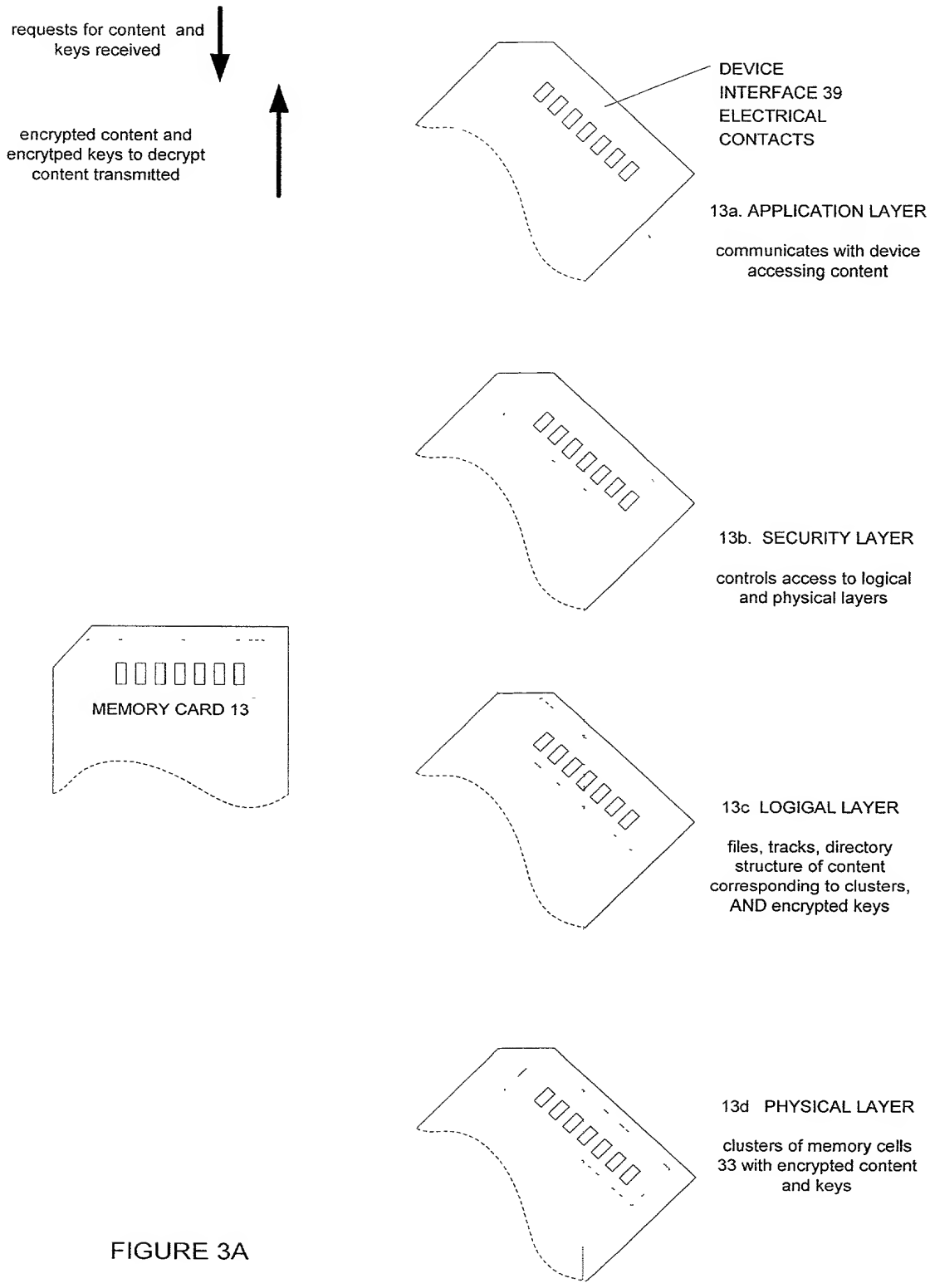

requests for content and keys received
encrypted content and encrytped keys to decrypt content transmitted
DEVICE INTERFACE 39 ELECTRICAL CONTACTS
13a. APPLICATION LAYER
communicates with device accessing content
13b. SECURITY LAYER
controls access to logical and physical layers
MEMORY CARD 13
13c LOGIGAL LAYER
files, tracks, directory structure of content corresponding to clusters, AND encrypted keys
13d PHYSICAL LAYER
clusters of memory cells 33 with encrypted content and keys

FIGURE 3A

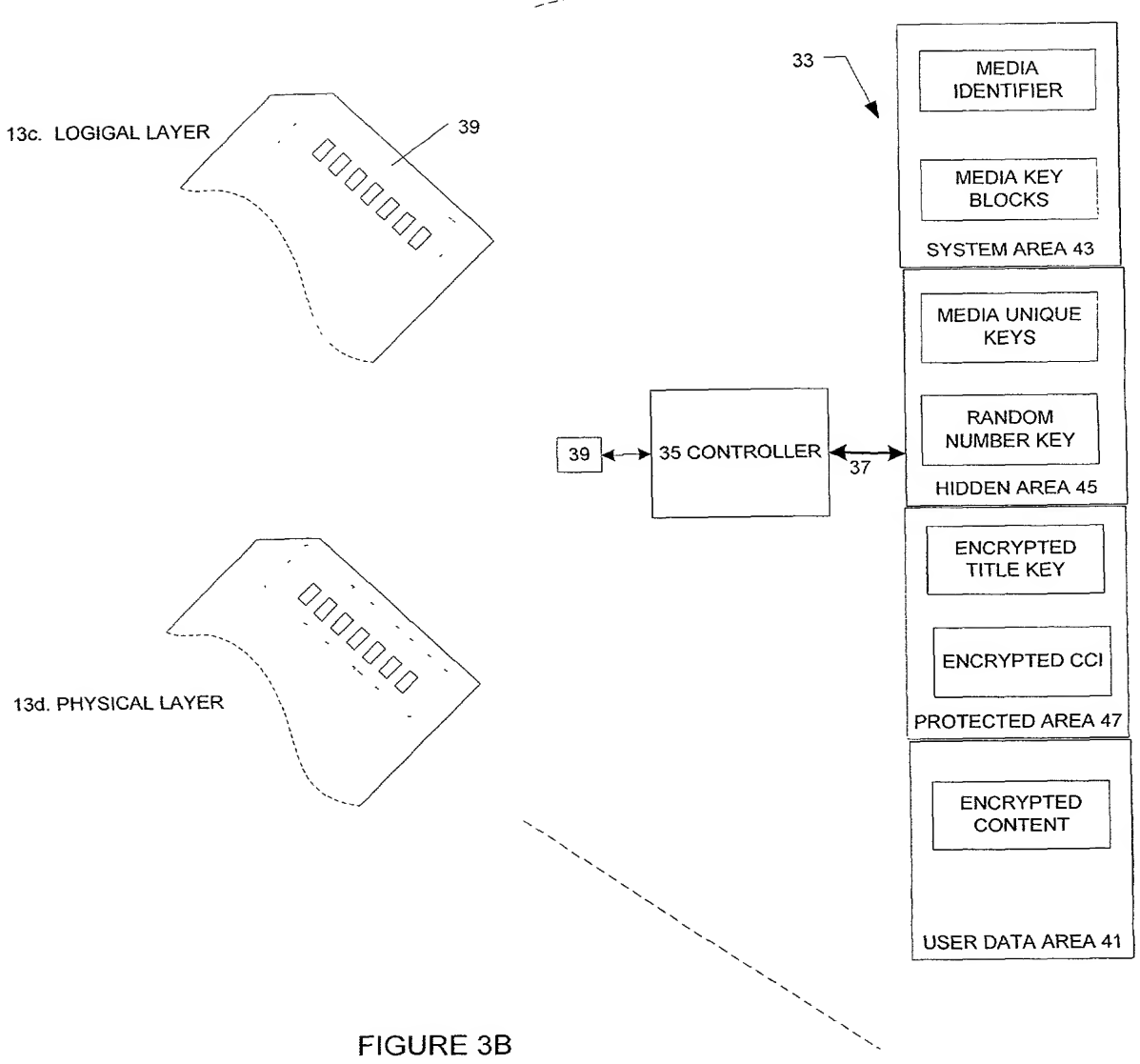

13c. LOGIGAL LAYER
39
13d. PHYSICAL LAYER
33
MEDIA IDENTIFIER
MEDIA KEY BLOCKS
SYSTEM AREA 43
MEDIA UNIQUE KEYS
RANDOM NUMBER KEY
HIDDEN AREA 45
39
35 CONTROLLER
37
ENCRYPTED TITLE KEY
ENCRYPTED CCI
PROTECTED AREA 47
ENCRYPTED CONTENT
USER DATA AREA 41

FIGURE 3B

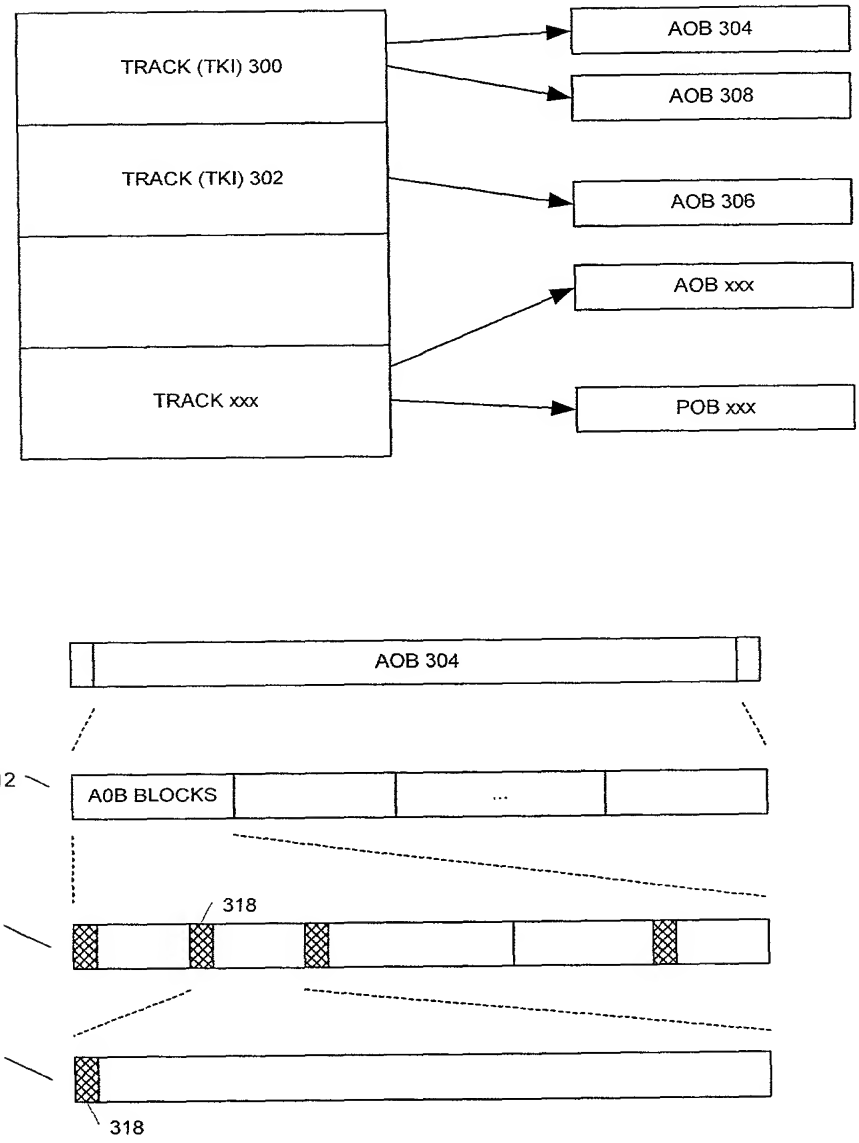

TRACK (TKI) 300
TRACK (TKI) 302
TRACK xxx
AOB 304
AOB 308
AOB 306
AOB xxx
POB xxx
AOB 304
312
A0B BLOCKS
...
318
316
320
318

FIG. 3C

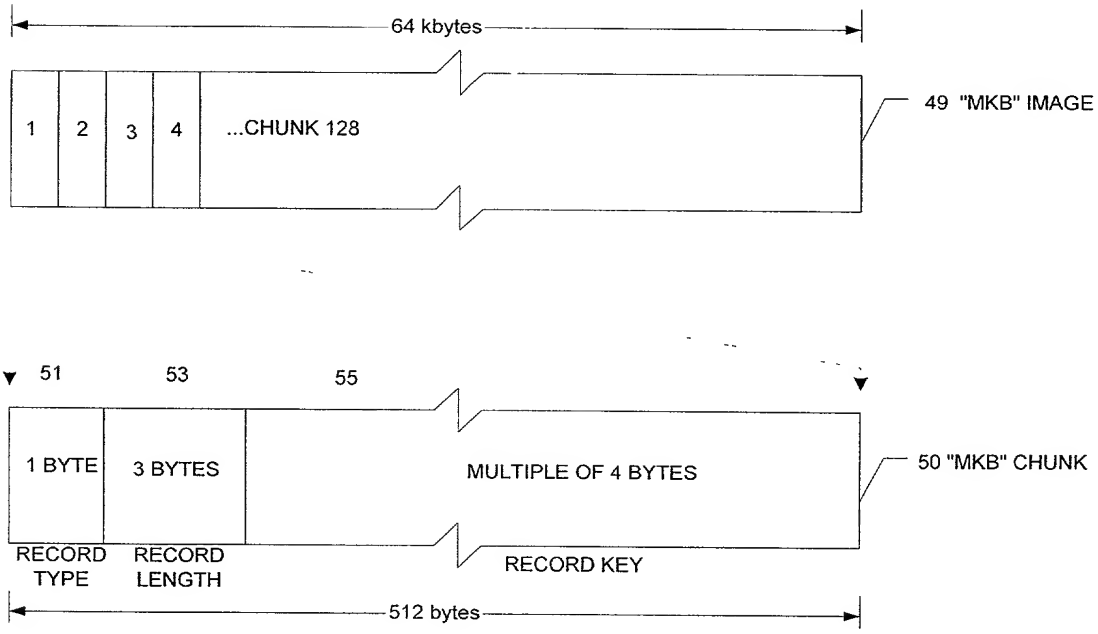

64 kbytes
1
2
3
4
...CHUNK 128
49 "MKB" IMAGE
51
53
55
1 BYTE
3 BYTES
MULTIPLE OF 4 BYTES
50 "MKB" CHUNK
RECORD TYPE
RECORD LENGTH
RECORD KEY
512 bytes

FIG. 4

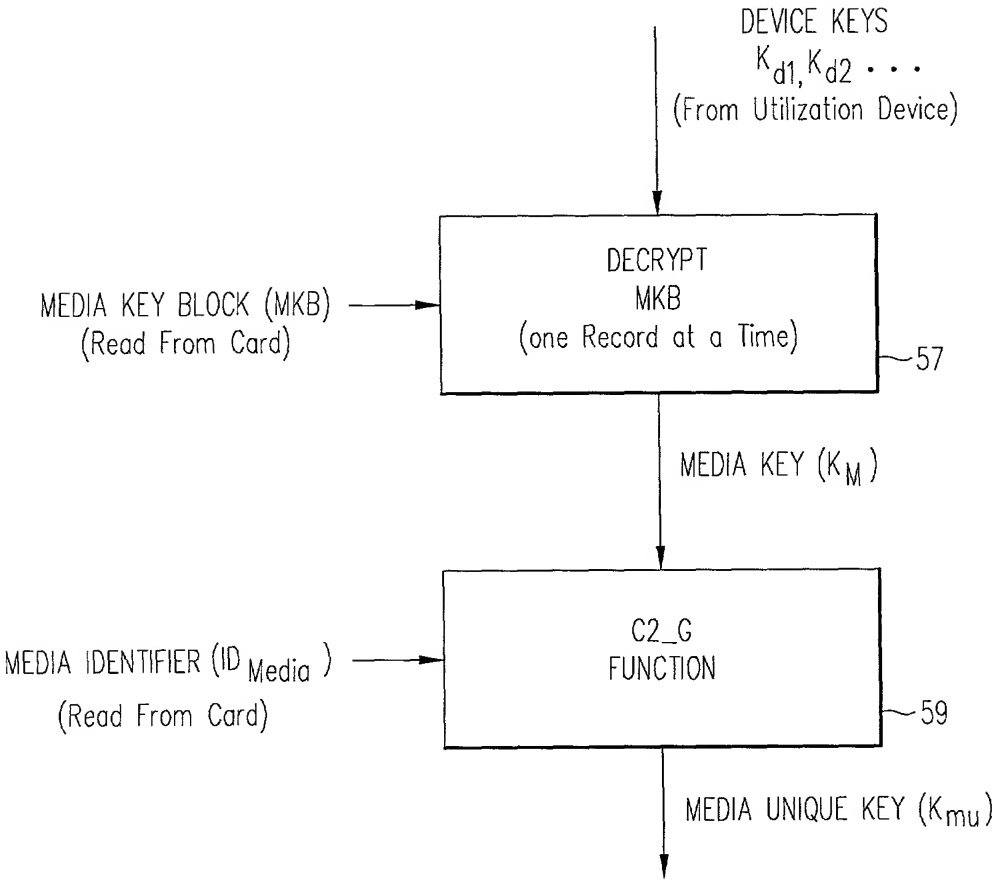

DEVICE KEYS
$K_{d1}, K_{d2} \cdots$
(From Utilization Device)
MEDIA KEY BLOCK (MKB)
(Read From Card)
DECRYPT
MKB
(one Record at a Time)
57
MEDIA KEY (K_M)
MEDIA IDENTIFIER (ID_{Media})
(Read From Card)
C2_G
FUNCTION
59
MEDIA UNIQUE KEY (K_{mu})

FIG. 5

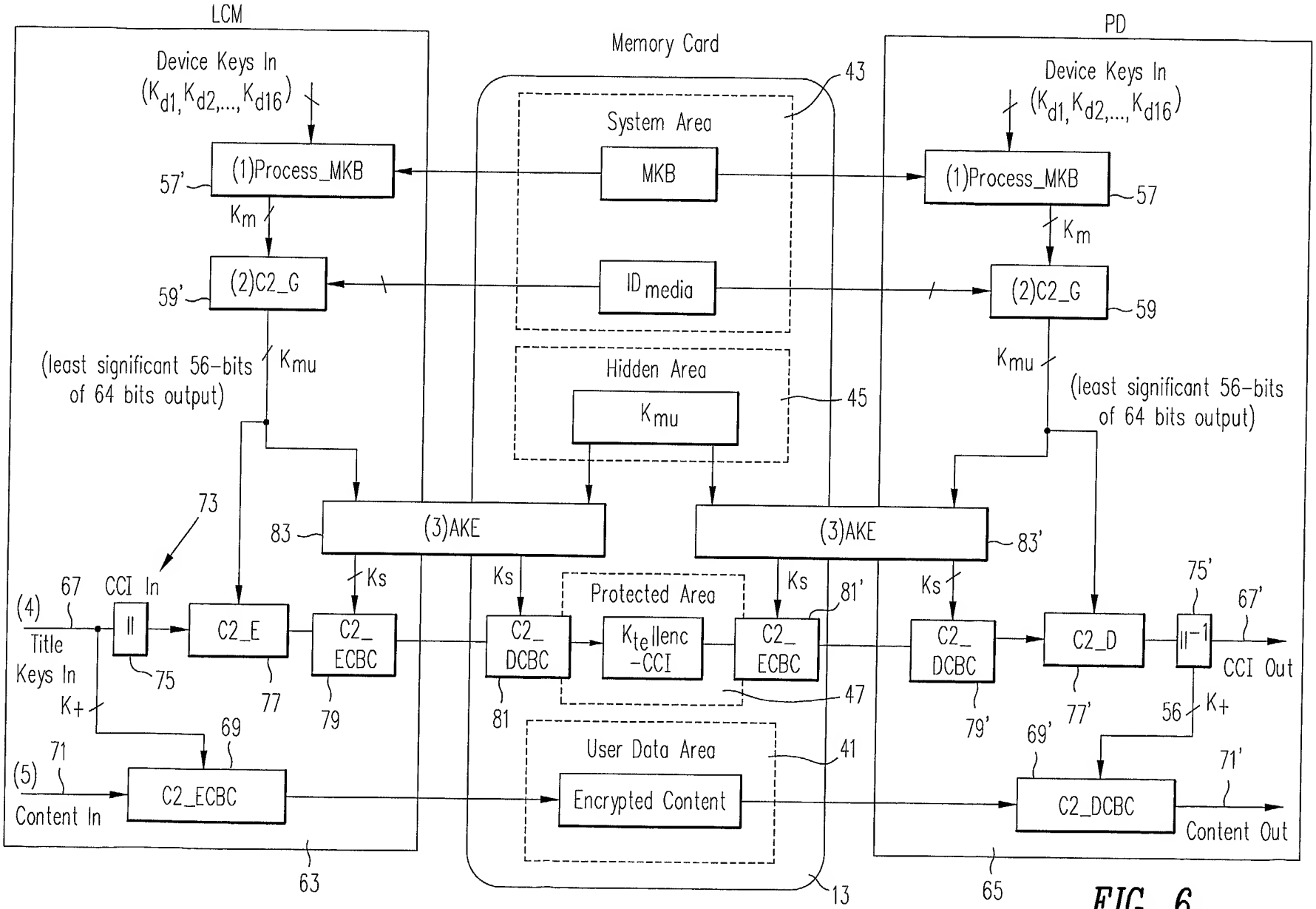

LCM
Memory Card
PD
Device Keys In
$(K_{d1}, K_{d2}, ..., K_{d16})$
(1)Process_MKB
57'
K_m
(2)C2_G
59'
(least significant 56-bits of 64 bits output)
K_{mu}
73
83
(3)AKE
CCI In
67
(4)
Title Keys In
K_+
II
75
C2_E
77
C2_ ECBC
79
Ks
(5)
71
Content In
69
C2_ECBC
63
43
System Area
MKB
ID media
Hidden Area
45
K_{mu}
Protected Area
C2_ DCBC
81
$K_{tellenc}$ -CCI
C2_ ECBC
81'
47
User Data Area
41
Encrypted Content
13
57
59
83'
75'
67'
CCI Out
C2_D
II^{-1}
56
79'
77'
69'
71'
C2_DCBC
Content Out
65

FIG. 6

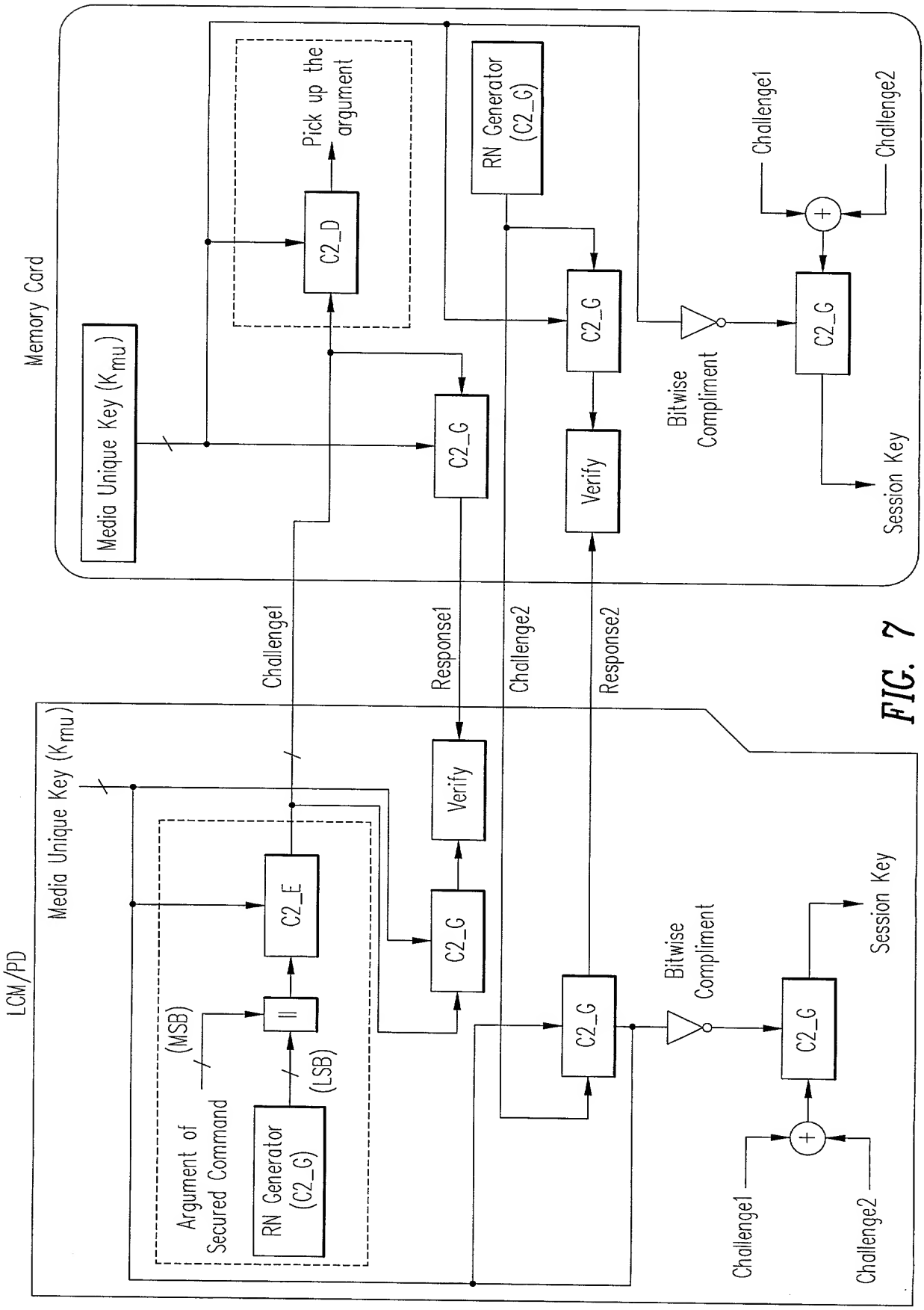

Memory Card
Media Unique Key (K_{mu})
C2_D
Pick up the argument
C2_G
RN Generator (C2_G)
C2_G
Verify
Bitwise Compliment
Challenge1
Challenge2
C2_G
Session Key
Challenge1
Response1
Challenge2
Response2
LCM/PD
Media Unique Key (K_{mu})
C2_E
(MSB)
(LSB)
Argument of Secured Command
RN Generator (C2_G)
C2_G
Verify
C2_G
Bitwise Compliment
C2_G
Session Key
Challenge1
Challenge2

FIG. 7

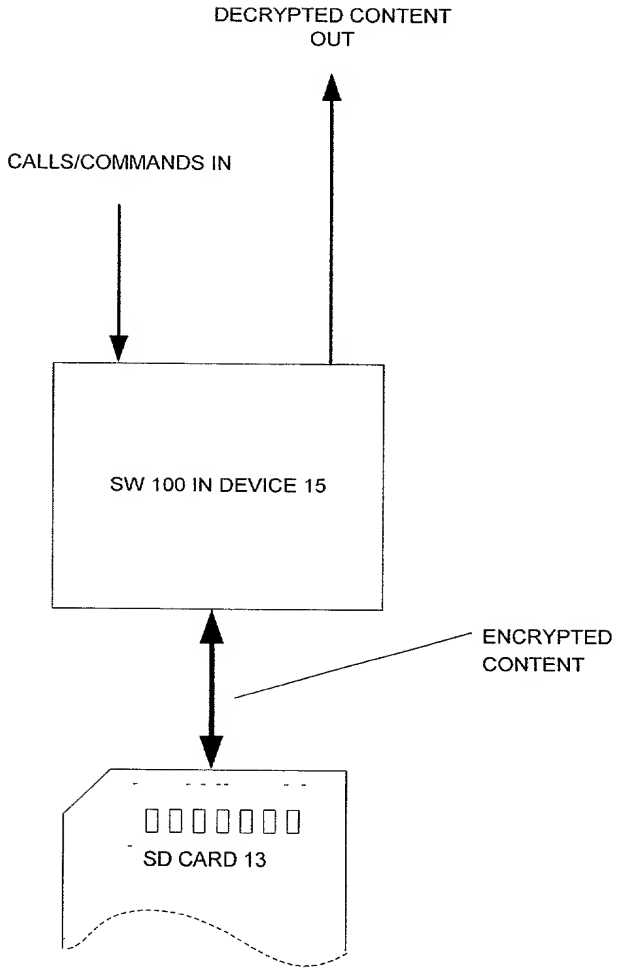

DECRYPTED CONTENT OUT
CALLS/COMMANDS IN
SW 100 IN DEVICE 15
ENCRYPTED CONTENT
SD CARD 13

FIG 8A

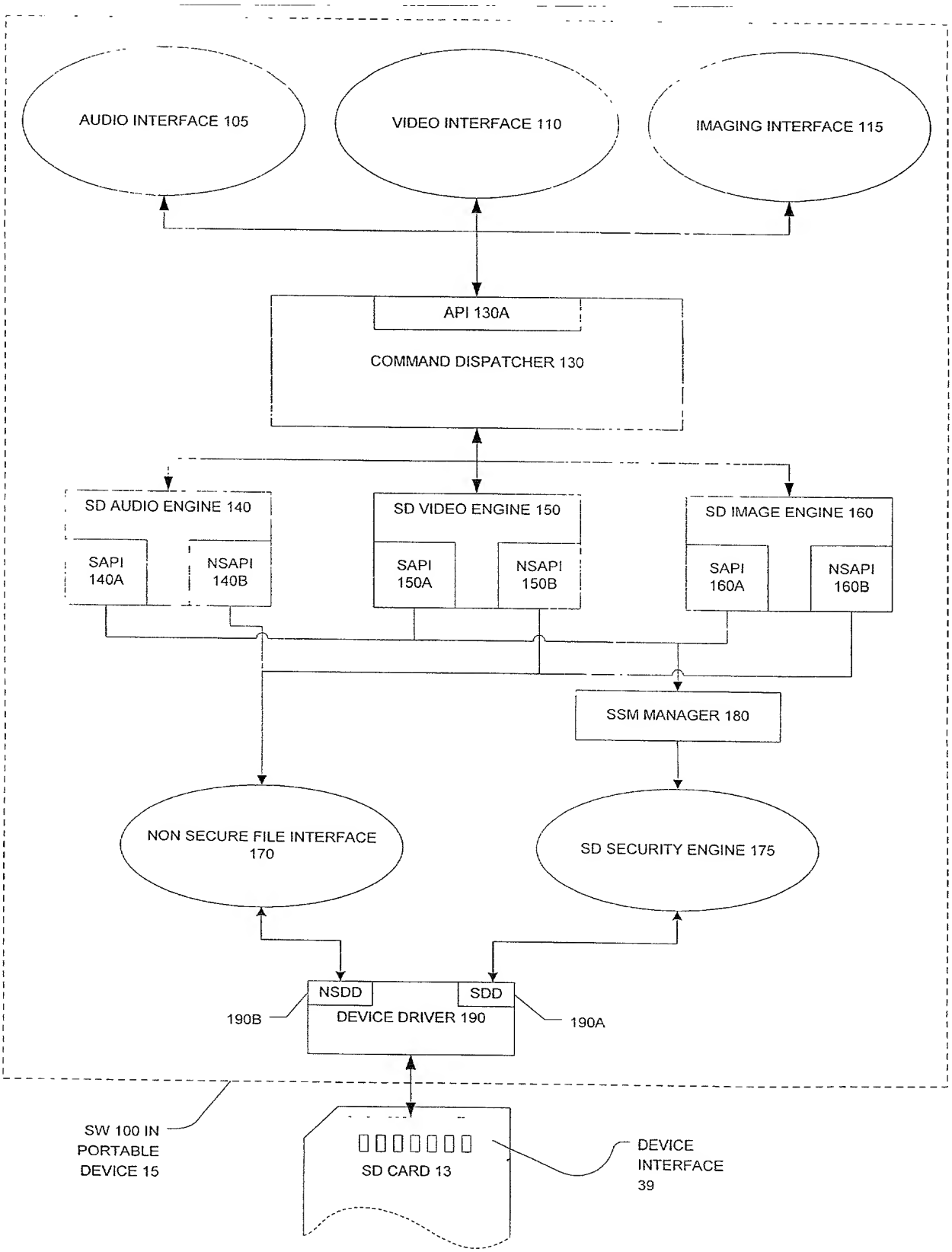

AUDIO INTERFACE 105
VIDEO INTERFACE 110
IMAGING INTERFACE 115
API 130A
COMMAND DISPATCHER 130
SD AUDIO ENGINE 140
SAPI 140A
NSAPI 140B
SD VIDEO ENGINE 150
SAPI 150A
NSAPI 150B
SD IMAGE ENGINE 160
SAPI 160A
NSAPI 160B
SSM MANAGER 180
NON SECURE FILE INTERFACE 170
SD SECURITY ENGINE 175
NSDD
SDD
DEVICE DRIVER 190
190B
190A
SW 100 IN PORTABLE DEVICE 15
SD CARD 13
DEVICE INTERFACE 39

FIGURE 8B

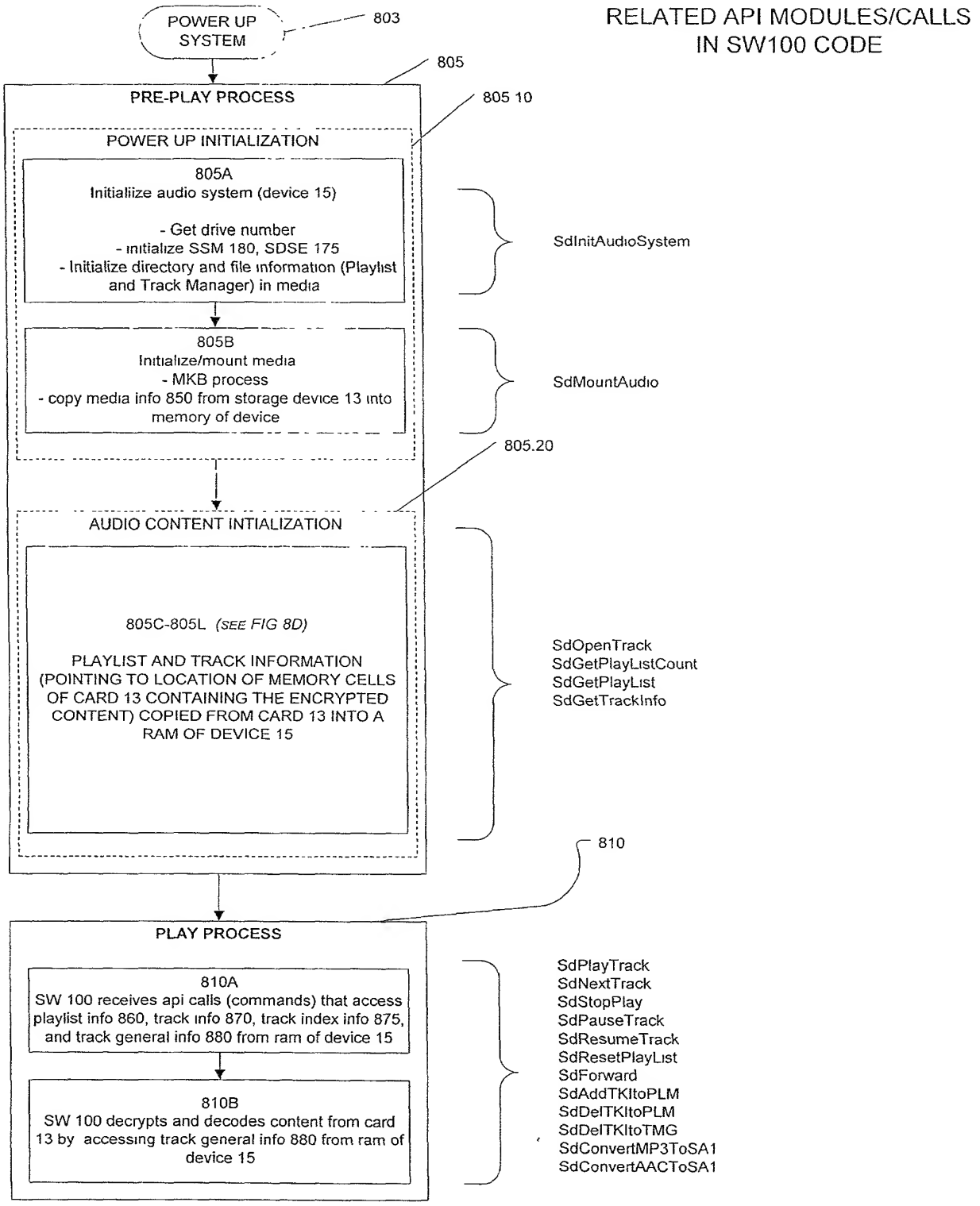

POWER UP SYSTEM
803
805
PRE-PLAY PROCESS
805 10
POWER UP INITIALIZATION
805A
Initialiize audio system (device 15)
- Get drive number
- initialize SSM 180, SDSE 175
- Initialize directory and file information (Playlist and Track Manager) in media
SdInitAudioSystem
805B
Initialize/mount media
- MKB process
- copy media info 850 from storage device 13 into memory of device
SdMountAudio
805.20
AUDIO CONTENT INTIALIZATION
805C-805L (SEE FIG 8D)
PLAYLIST AND TRACK INFORMATION (POINTING TO LOCATION OF MEMORY CELLS OF CARD 13 CONTAINING THE ENCRYPTED CONTENT) COPIED FROM CARD 13 INTO A RAM OF DEVICE 15
SdOpenTrack
SdGetPlayListCount
SdGetPlayList
SdGetTrackInfo
810
PLAY PROCESS
810A
SW 100 receives api calls (commands) that access playlist info 860, track info 870, track index info 875, and track general info 880 from ram of device 15
810B
SW 100 decrypts and decodes content from card 13 by accessing track general info 880 from ram of device 15
SdPlayTrack
SdNextTrack
SdStopPlay
SdPauseTrack
SdResumeTrack
SdResetPlayList
SdForward
SdAddTKItoPLM
SdDelTKItoPLM
SdDelTKItoTMG
SdConvertMP3ToSA1
SdConvertAACToSA1
RELATED API MODULES/CALLS IN SW100 CODE

FIG 8C

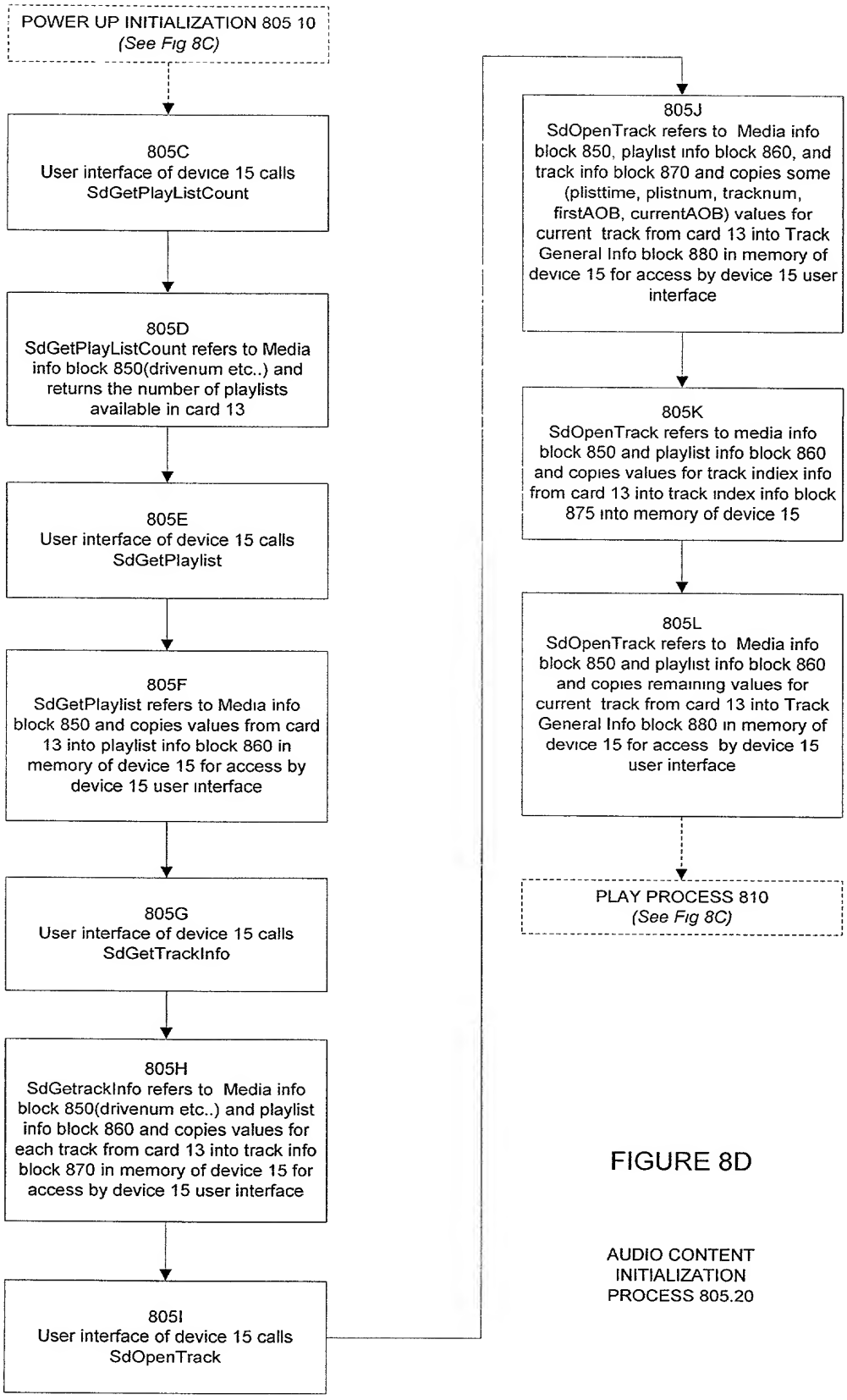
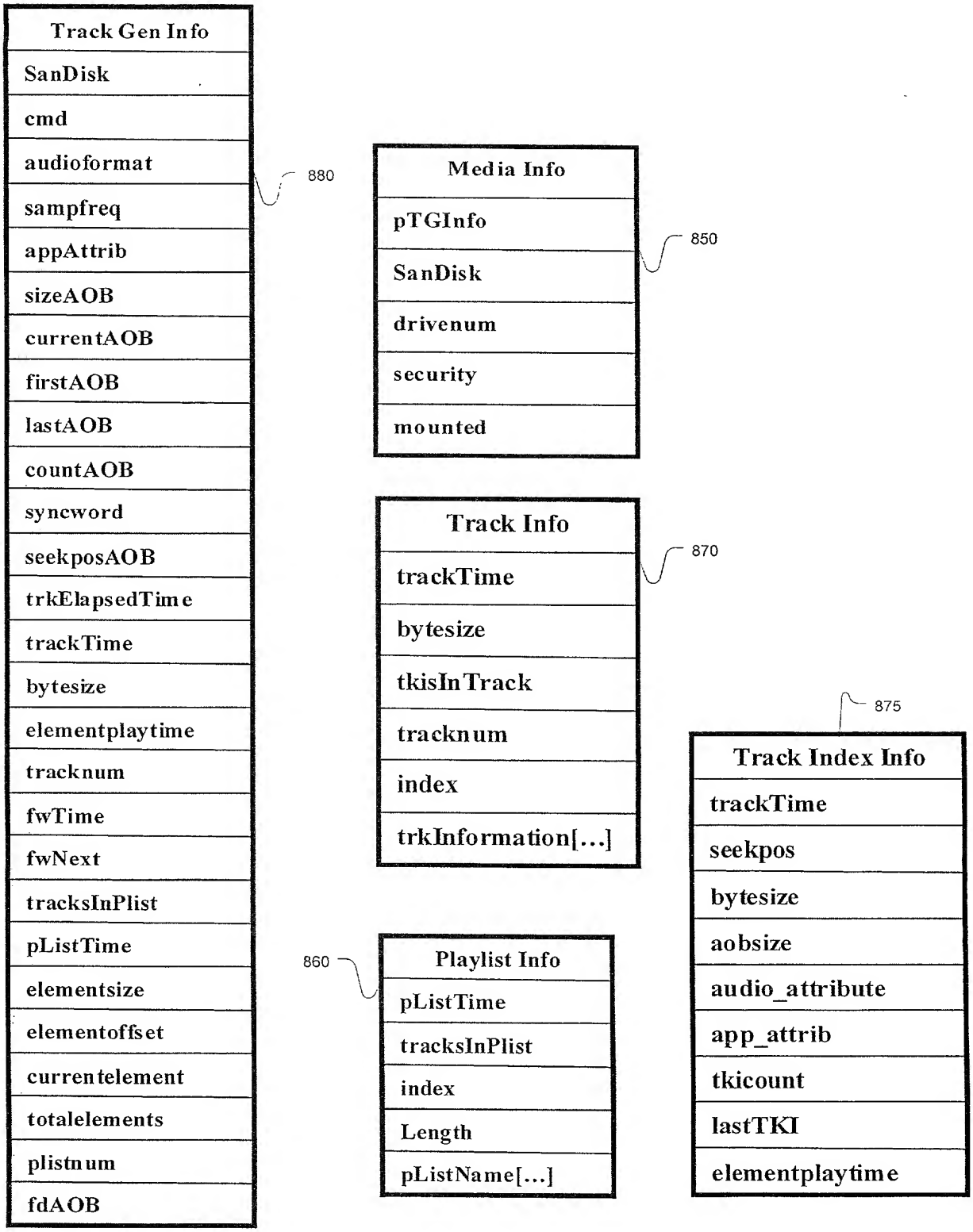


FIGURE 8D

AUDIO CONTENT
INITIALIZATION
PROCESS 805.20



PRE-PLAY STRUCTURES
FIG 8E

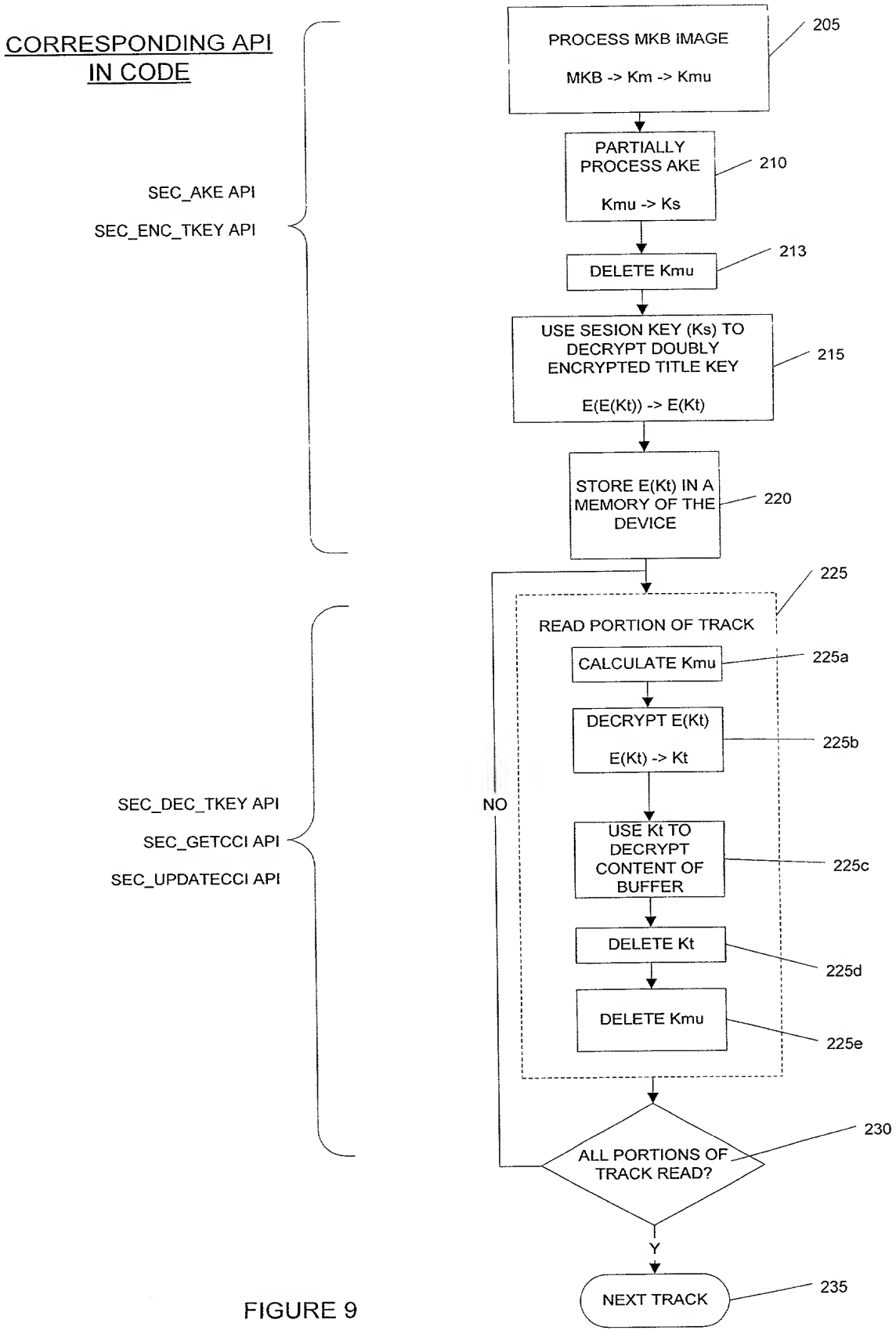

CORRESPONDING API IN CODE
SEC_AKE API
SEC_ENC_TKEY API
SEC_DEC_TKEY API
SEC_GETCCI API
SEC_UPDATECCI API
PROCESS MKB IMAGE
MKB -> Km -> Kmu
205
PARTIALLY PROCESS AKE
Kmu -> Ks
210
DELETE Kmu
213
USE SESION KEY (Ks) TO DECRYPT DOUBLY ENCRYPTED TITLE KEY
E(E(Kt)) -> E(Kt)
215
STORE E(Kt) IN A MEMORY OF THE DEVICE
220
225
READ PORTION OF TRACK
CALCULATE Kmu
225a
DECRYPT E(Kt)
E(Kt) -> Kt
225b
USE Kt TO DECRYPT CONTENT OF BUFFER
225c
DELETE Kt
225d
DELETE Kmu
225e
NO
ALL PORTIONS OF TRACK READ?
230
Y
NEXT TRACK
235

FIGURE 9

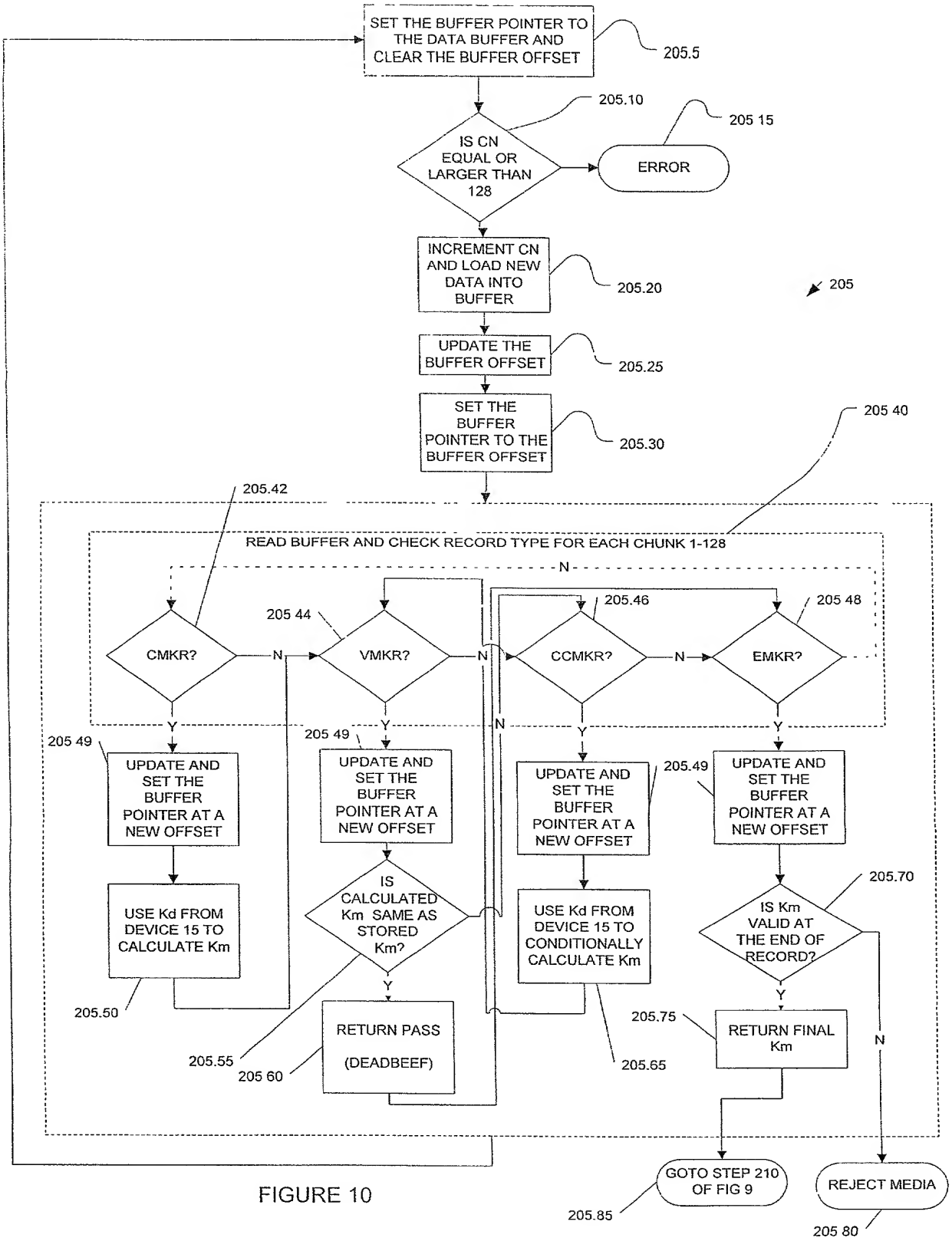

SET THE BUFFER POINTER TO THE DATA BUFFER AND CLEAR THE BUFFER OFFSET
205.5
205.10
IS CN EQUAL OR LARGER THAN 128
205 15
ERROR
INCREMENT CN AND LOAD NEW DATA INTO BUFFER
205.20
205
UPDATE THE BUFFER OFFSET
205.25
SET THE BUFFER POINTER TO THE BUFFER OFFSET
205.30
205 40
205.42
READ BUFFER AND CHECK RECORD TYPE FOR EACH CHUNK 1-128
CMKR?
VMKR?
205 44
CCMKR?
205.46
EMKR?
205 48
N
Y
205 49
UPDATE AND SET THE BUFFER POINTER AT A NEW OFFSET
USE Kd FROM DEVICE 15 TO CALCULATE Km
205.50
IS CALCULATED Km SAME AS STORED Km?
205.55
RETURN PASS (DEADBEEF)
205 60
USE Kd FROM DEVICE 15 TO CONDITIONALLY CALCULATE Km
205.65
205.70
IS Km VALID AT THE END OF RECORD?
205.75
RETURN FINAL Km
GOTO STEP 210 OF FIG 9
205.85
REJECT MEDIA
205 80

FIGURE 10